UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

| APPLICATION NO. | FILING DATE | FIRST NAMED INVENTOR | ATTORNEY DOCKET NO. | CONFIRMATION NO. |
|---|---|---|---|---|
| 09/680,126 | 10/03/2000 | John Newlin | 083818 0269851 | 6144 |

27498 7590 12/02/2004

PILLSBURY WINTHROP LLP
2475 HANOVER STREET
PALO ALTO, CA 94304-1114

| EXAMINER | |
|---|---|
| STEELMAN, MARY J | |
| **ART UNIT** | **PAPER NUMBER** |
| 2122 | |

DATE MAILED: 12/02/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

| Office Action Summary | Application No. | Applicant(s) | |
|---|---|---|---|
| | 09/680,126 | NEWLIN ET AL. | |
| | Examiner | Art Unit | |
| | Mary J. Steelman | 2122 | |

*-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --*

**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

1)☒ Responsive to communication(s) filed on *3/8/2004 & 8/24/2004*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

4)☒ Claim(s) *1-20* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-20* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

**Application Papers**

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *03 October 2000* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

## DETAILED ACTION

1. This action is responsive to RCE received 24 August 2004. Amendments and remarks received 8 March 2004 are addressed in this response. Per Applicant's request, claims 9 and 12 have been amended. Claims 1-20 are pending.

### *Claim Rejections - 35 USC § 112*

2. In view of the amendment to claim 9, the prior 35 USC 112 first paragraph rejection is hereby withdrawn.

### *Claim Rejections - 35 USC § 103*

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

4. Claims 1-20 are rejected under 35 U.S.C. 103(a) as being unpatentable over US Patent 6,587,995 to Duboc et al.

Duboc disclosed (Abstract) "An apparatus, program product and method incorporate into an enhanced programmable core model an embedded debug monitor to provide integrated graphical debugging functionality" Col. 3, lines 26-28, "...a significant need exists in the art for a manner of supporting debugging functionality for use in conjunction with simulation of a programmable core."

Per claims 1 and 13:

-transmitting, using a debugger, a state-accessing instruction stream to an interpreting agent, the interpreting agent being capable of interpreting that stream;

(See fig. 3, #44 (debug GUI) and #48 (interpreter) & fig. 9, #208 (capture register values / state accessing instruction stream), and col. 7, lines 37-45, "User interface with the debug monitor is provided through a debug GUI script executed by a script interpreter. Communication (transmitting state accessing instruction stream) between the debug GUI script and debug monitor is provided through a communications channel, with debug parameters used to configure the debug monitor transmitted from the debug GUI script to the debug monitor...)

-causing, using the state-accessing instruction stream, the interpreting agent to return the state of the processor to the debugger.

(See fig. 3, #44 (debug GUI) and #48 (interpreter) & fig. 9, #208 (capture register values / state accessing instruction stream), and col. 7, lines 37-45, "...with results of the debug operations transmitted by the debug monitor to the script for display to the user(return the state of the processor to the debugger).")

Duboc disclosed transmitting communications and debug parameters to a debug monitor. He did not specifically disclose that the communications were 'state-accessing instruction streams'. A script is executed by a script interpreter (col. 7, line 38). Communications (transmitted) cause the debug monitor to (col. 7, line 36) "**monitor the state** (state-accessing instruction stream) of

the programmable core model". (emphasis added) The (col. 7, lines 53-55) "enhanced core model is typically implemented in a hardware description language..."

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to recognize that Duboc's invention using an interpreter, a debugger, and a communication process to access the state of a processor to be broadly interpreted to meet the limitations of claims1 and 13, as both use the same features to accomplish to same results. Additionally note that Duboc disclosed (col. 8, lines 26-29, "It will be appreciated that a wide variety of alternate platforms and development tools may be utilized for implementing an enhanced processor core model consistent with the invention" and col. 8, lines 52-54, "...functions allocated between the debug GUI script and the debug monitor may alternately be allocated to different extents in each of the two components." Thus Duboc suggested that slight variations of his invention were also implied.

Per claims 2, 6, 14, and 18:

-interpreting agent is a monitor program.

(Col. 7, line 42 "debug monitor" & col. 13, lines 37-40, "The specific register contents to be monitored..." Agent performs the monitoring process.)

Per claims 3, 7, 15 and 19:

-interpreting agent is an instruction insertion server.

(See fig. 1, "networked server" and fig. 2. Col. 5, lines 31-44 & 53-59, "Fig. 1 illustrates a computer system for use in developing and/or testing an enhanced programmable core model...networked computer system...coupled to server..." and "...apparatus may represent practically any type of computer, computer system or other programmable electronic device including...a server computer..." Also, col. 6, lines 15-16, "Computer also typically receives a number of inputs and outputs for communicating information externally...")

Per claims 4, 8, 16, and 20:

-interpreting agent is an architectural simulator.

(Col. 7, lines 48-49, "...simulating the functions of a core model and any integrated circuit design incorporating the same...", col. 7, lines 63-64, "...encapsulate the definition in a simulator-specific wrapper...", and col. 8, lines 4-6, "...a collation phase, where components defined in the form of drop ins are collected to represent a particular core model.")

Per claim 5:

-reading, using the debugger, information describing the configurable processor's state architecture;

(Col. 16, lines 48-52, "Next a debug monitor developed in accordance with the design flow described above in connection with Fig. 10 is provided along with the translated netlist to a model generation tool to generate an enhanced programmable core model consistent with the invention...")

-generating, using the debugger, the instruction stream based on the information.

(Col. 16, lines 59-67, "The simulator-specific programmable core model includes...netlist and debug monitor encapsulated in a hardware description language wrapper, as appropriate for the particular simulator to which the core model is directed. Wrapper describes the inputs and outputs of the core block and calls the compiled netlist block and debug monitor as hierarchical child blocks to permit the core model to be integrated...")

Per claim 9:

-a debugger library for automatically generating information necessary to describe save and restore instructions for state of the configurable processor base on the user description.

(See fig. 10. Col. 14, line 53- col. 15, line 6, "To develop an enhanced processor core...From this data, both the debug monitor and debug GUI script program code is developed. Typically the debug monitor and script is based upon customizing and/or reusing generic library program code (save and restore instructions for state)...")

Duboc suggested "software for automatically generating a hardware description of a configurable processor from a user description of that processor" but failed to provided details regarding 'automatically generating.'

(Col. 4, lines 28-29, "...development of an enhanced programmable core model...", col. 4, lines 54-56, "...a core model incorporates functional information formatted in a hardware description language such as VHDL (automatically generating a hardware description)...", col. 4, lines 61-62, "...some synthesis information may be incorporated into an enhanced programmable core

model...", col. 5, lines 30-33, "...FIG. 1 illustrates a computer system for use in developing and/or testing an enhanced programmable core model...", col. 6, lines 61-64, "...the programs may represent logical or functional representations of an integrated circuit design, i.e., a functional model of an integrated circuit design suitable for interaction with a simulation tool", col. 8, lines 10-11, "...generate a simulator specific programmable core model")

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to modify Duboc's invention to more clearly provide for "automatically generating a hardware description of a configurable processor from a user description of that processor" because Duboc suggested that the netlist is used, and some synthesis information may be incorporated, to develop the core model. Also see col. 17, lines 8-16, "It will be appreciated that the development of an enhanced programmable core model in the manner discussed herein using the IPGuard and ModelGen tools **would be well within the ability of one of ordinary skill in the art** having the benefit of the instant disclosure. Moreover, other tools may be utilized to generate an enhanced programmable core model consistent with the invention. Therefore, the invention is not limited to the specific implementation disclosed herein." (emphasis added)

Per claim 10:

-reading a description of save and restore state information of a configurable processor;

-generating saving and restoring state instruction streams based on the description.

(Col. 15, lines 36-59, "With respect to the registers defined in the processor specification, a register window display script is built (generating)...and is configured to receive register state

information over the register pipe and display such information (reading description of save and restore state information)...")

Per claim 11:

-identifying interdependencies in state;

-generating a complete and correct save and restore sequence based on the interdependencies.

(Col. 14, line 53 – col.15, line 6 also col. 16, lines 22-26 & 32-39, "From the standpoint of a programmable core developer, it may be desirable to provide either generic cores...or...simulator-specific programmable cores..." & "Any program core developer typically starts with a core netlist..." )

Per claim 12:

-means for retrieving system topology information of a chip containing multiple cores from a computer-readable file;

(Col. 6, line 2-3, "Processor may represent one or more processors (multiple cores)..."

-means for determining where elements are in a system described by the file

(Col. 16, lines 32-33, 48-52, & 63-67, "...start with a core netlist...", "...translated netlist to a model generation tool...", "Wrapper describes the inputs and outputs of the core block (determining where elements are in a system)..." )

Duboc disclosed "means responsive to the determining means for directing … to an appropriate one of the multiple cores", but failed to specify that the communications were 'a state-accessing instruction stream'.

(Col. 8, lines 12-25, "To implement graphical debugging functionality within a programmable core model…the debug monitor is defined…and configured…suitable for collation with a processor core model…The debug monitor typically includes a plurality of debug operation handlers that are invoked (means for directing state accessing instruction stream)…essentially watches the states of the various nodes…and captures relevant information with regard to the state of various signals…" Duboc suggested (col. 9, lines 45-47), "…multiple processor core (multiple cores) models may be simulated concurrently in a multi-processor integrated circuit design…" )

Duboc did disclose that communications (state-accessing instruction stream) were delivered to monitor the processor core and result in a return of the detected state. See FIG. 3 and col. 7, lines 29-45.

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to modify Duboc's invention to suggest that "communications" delivered to the processor core to monitor and return state information are interpreted as 'state-accessing instruction streams', because they accomplish the same result.

Per claim 17:

-read information describing the configurable processor's state architecture;

(Col. 7, line 61-col. 8, line 11, "...ModelGen tool is used to take a C-language description of a generic model, typically in the form of a C-compiled netlist, and encapsulate the definition...The ModelGen tool operates in two phases, a collation phase, where components defined in the form of drop ins are collected to represent a particular core model..." ,col. 16, lines 32-33, 48-52, & 63-67, "...start with a core netlist...", "...translated netlist to a model generation tool...", "Wrapper describes the inputs and outputs of the core block..." and col. 17, lines 6-7, "The resulting enhanced programmable core model..." The netlist defines the architecture.)

-generate the instruction stream based on the information.

(Col. 8, lines 12-25, "To implement graphical debugging functionality within a programmable core model...the debug monitor is defined...and configured (instruction stream generated)...suitable for collation with a processor core model (instruction stream based on the information)...The debug monitor typically includes a plurality of debug operation handlers that are invoked (generate instruction stream)...essentially watches the states of the various nodes...and captures relevant information with regard to the state of various signals..." )

***Response to Arguments***

5. Applicant has argued, in substance, the following:

(A) As Applicant has pointed out on page 6, 4th paragraph, of Remarks, received 8 March 2004, "Duboc does not describe or suggest transmitting anything from a debugger to an interpreting agent, as required by claims 1 and 13.

Examiner's Response:

Claim language does not recite "from a debugger", but merely recites "using a debugger". A debugger is used, communication is transmitted, to an interpreting agent. See FIG. 3 and response to claim 1 above. Additionally, it is unclear from the claim language of claim 1 that the debugger is remotely located or otherwise located in a separate/different system, as suggested in the remarks.

(B) As Applicant has pointed out on page 6, 5th paragraph, of Remarks, "...nothing in Duboc describes or suggests transmitting a state-accessing instruction stream to an interpreting agent, which is then used to return processor state information to the debugging application..."

Examiner's Response:

State accessing instruction stream is transmitted and state returned through communication channel. See FIG. 3 and response to claim 1 above.

(C) As Applicant has pointed out on page 7, 1st paragraph, of Remarks, "Nothing in Duboc discloses or suggests anything about including actual processor instructions in a "debug GUI

script," much less instructions that are executed by the processor to access state, as required by independent claims 1 and 13."

Examiner's Response:

A debug GUI script is not a claim limitation. Communications (instructions) are transmitted to access state. See FIG. 3.

(D) As Applicant has pointed out on page 7, 4th paragraph, of Remarks, "Duboc does not disclose or suggest anything about configurable processors, much less generating a hardware description of a configurable processor based on a user description."

Examiner's Response:

See response to claim 9 above.

***Conclusion***

6. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Mary Steelman, whose telephone number is (571) 272-3704. The examiner can normally be reached Monday through Thursday, from 7:00 AM to 5:30 PM If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan

Q. Dam can be reached at (571) 272-3695. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Mary Steelman

11/10/2004

JOHN CHAVIS
PATENT EXAMINER
ART UNIT 2124